TOWARD ROBUST LEARNING OF THE GAUSSIAN MIXTURE STATE EMISSION DENSITIES FOR HIDDEN MARKOV MODELS

Hao Tang, Mark Hasegawa-Johnson, Thomas S. Huang

University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering
405 North Mathews Avenue, Urbana, IL 61801 USA

ABSTRACT

One important class of state emission densities of the hidden Markov model (HMM) is the Gaussian mixture densities. The classical Baum-Welch algorithm often fails to reliably learn the Gaussian mixture densities when there is insufficient training data, due to the large number of free parameters present in the model. In this paper, we propose a novel strategy for robustly and accurately learning the Gaussian mixture state emission densities of the HMM. The strategy is based on an ensemble framework for probability density estimation in which the learning of the Gaussian mixture densities is formulated as a gradient descent search in a function space. The resulting learning algorithm is called "the boosting Baum-Welch algorithm." Our preliminary experiment results on emotion recognition from speech show that the proposed algorithm outperforms the original Baum-Welch algorithm on this task.

Index Terms— hidden Markov model, Gaussian mixture density, Baum-Welch algorithm, boosting

1. INTRODUCTION

The hidden Markov model (HMM) [1] is a doubly stochastic process consisting of an underlying, hidden, discrete random process which possesses the Markov property (namely a Markov chain) and an observed, discrete, continuous, or mixed discrete-continuous random process which is a probabilistic function of the underlying Markov chain. The HMM is a natural probabilistic model for describing sequential data such as speech signals (time series), DNA sequences, character strings or sentences in a language, and so forth [2]. It is a generative model by which we can characterize and evaluate the properties of many data sources without having actual access to them. In practice, the HMM has been widely applied to a variety of applications including automatic speech recognition [3], natural language processing [4], handwriting recognition [5], etc.

The three well-known basic problems associated with the HMM are the evaluation of the observation likelihood, the determination of the optimal state sequence, and the learning of the model parameters [1]. The classical Baum-Welch algorithm [6], aimed at solving the model parameter learning problem, can give accurate parameter estimates for the discrete-observation HMM. For the continuous-observation HMM, one important class of state emission densities is the Gaussian mixture densities [7]. The Baum-Welch algorithm often fails to reliably learn the Gaussian mixture densities when there is insufficient training data, due to the large number of free parameters present in the model. For instance, if an HMM has N states, and each state is modeled by an M-component Gaussian mixture density, the total number of model parameters is $N\{(M-1)+M[D+D(D+1)/2]\}+K$, where D is the dimension of the observation vectors, and K is the number of model parameters governing the initial state probabilities and state transition probabilities of the HMM. The "curse of dimensionality" [8] implies that, unless the number of observation vectors available for training the HMM is at least multiple (e.g. $5 \sim 10$) times that of model parameters, the Baum-Welch algorithm can give rather poor parameter estimates for the model. Unfortunately, we always face situations where there is insufficient training data in all kinds of real-world application.

The poor parameter estimates are primarily attributed to the fact that the Baum-Welch algorithm cannot reliably learn the Gaussian mixture state emission densities of the HMM due to data sparsity. To alleviate this problem, we propose a novel strategy to improve the robustness and accuracy of the Baum-Welch algorithm for learning the Gaussian mixture state emission densities of the HMM. The strategy is based on an ensemble or boosting framework [9] for probability density estimation in which the learning of the Gaussian mixture densities is formulated as a gradient descent search in a function space. The resulting learning algorithm is called "the boosting Baum-Welch algorithm."

We apply the proposed boosting Baum-Welch algorithm to the problem of emotion recognition from speech [10] based on a speech database consisting of 720 English short sentences each of which was spoken by an American female speaker in the neutral, happy, sad, and anger manners, respectively. Our preliminary experiment results clearly show that the proposed algorithm indeed improves the learning accuracy of the Gaussian mixture state emission densities of the emotion category dependent HMMs, leading to better emotion recognition performance than the original Baum-Welch algorithm.

This paper is organized as follows. Section 2 briefly reviews the HMM and Baum-Welch algorithm. Section 3 describes the proposed boosting Baum-Welch algorithm in great detail. Section 4 presents our experiment results, followed by conclusions in Section 5.

2. THE BAUM-WELCH ALGORITHM

In the mathematical language, we may completely specify an HMM by its parameters $\lambda = \{A, B, \Pi\}$, where A is the state transition probability matrix whose entries, $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$, $1 \leq i, j \leq N$, determine the probabilities of transition from state S_i to state S_j at time t, B is the state emission probability matrix whose entries, $b_{jk} = P(o_t = v_k | q_t = S_j)$, $1 \leq j \leq N, 1 \leq k \leq M$, determine the probabilities of emitting an observation symbol v_k given that the model is in state S_j at time t, and Π is the initial state probability matrix whose entries, $\pi_i = P(q_1 = S_i)$,

This work was supported in part by the Grant NSF 08-03219.

$1 \leq i \leq N$, determine the probabilities of the model being initially in state S_i. For the case of the continuous-observation HMM, the entries of B are given by continuous probability density functions, namely $b_j(o_t) = P(o_t|q_t = S_j)$, $1 \leq j \leq N$. One important class of continuous probability density functions widely used for the state emission densities of the continuous-observation HMM is the Gaussian mixture density functions of the form

$$\begin{aligned} b_j(o_t) &= \sum_{k=1}^{M} c_{jk} N(o_t|\mu_{jk}, \Sigma_{jk}) \\ & \quad 1 \leq j \leq N, 1 \leq k \leq M \end{aligned} \tag{1}$$

where M is the number of Gaussian components, c_{jk} is the k^{th} component weight, and $N(o_t|\mu_{jk}, \Sigma_{jk})$ is a multivariate Gaussian density function with mean vector μ_{jk} and covariance matrix Σ_{jk}.

The Baum-Welch algorithm learns the parameters of an HMM, λ, given an observation sequence, $O = o_1 o_2 \cdots o_T$. It is an iterative re-estimation procedure that increases the log likelihood $P(O|\lambda)$ monotonically, and is based on an efficient algorithm known as the forward-backward algorithm. A "forward" variable, $\alpha_t(i)$, defined as the probability of the partial observation sequence up to time t and state S_i being occupied at time t, is given by

$$\begin{aligned} \alpha_t(i) &= P(o_1 o_2 \cdots o_t, q_t = S_i|\lambda) \\ & \quad 1 \leq i \leq N, 1 \leq t \leq T \end{aligned} \tag{2}$$

The α's are computed iteratively as follows:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad i = 1, 2, \cdots, N \tag{3}$$

$$\begin{aligned} \alpha_t(j) &= \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(o_t) \\ & \quad j = 1, 2, \cdots, N, t = 2, 3, \cdots, T \end{aligned} \tag{4}$$

A "backward" variable, $\beta_t(i)$, defined as the probability of the partial observation sequence from time $t+1$ to T given that the state S_i is occupied at time t, is given by

$$\begin{aligned} \beta_t(i) &= P(o_{t+1} o_{t+2} \cdots o_T|q_t = S_i, \lambda) \\ & \quad 1 \leq i \leq N, 1 \leq t \leq T \end{aligned} \tag{5}$$

Likewise, the β's are computed iteratively as follows:

$$\beta_T(j) = 1, \quad j = 1, 2, \cdots, N \tag{6}$$

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \\ & \quad i = 1, 2, \cdots, N, t = T-1, T-2, \cdots, 1 \end{aligned} \tag{7}$$

The Baum-Welch algorithm starts with random (or smarter) initialization of the model parameters λ and proceeds as follows:

(1) Re-estimation of the new model parameters $\hat{\lambda}$:

$$\hat{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N} \alpha_1(j)\beta_1(j)} \tag{8}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \tag{9}$$

$$\hat{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \tag{10}$$

$$\hat{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) o_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{11}$$

$$\hat{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)(o_t - \hat{\mu}_{jk})(o_t - \hat{\mu}_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{12}$$

where $1 \leq i, j \leq N, 1 \leq k \leq M$, and

$$\gamma_t(j,k) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \frac{c_{jk} N(o_t|\mu_{jk}, \Sigma_{jk})}{\sum_{m=1}^{M} c_{jm} N(o_t|\mu_{jm}, \Sigma_{jm})} \tag{13}$$

(2) Test of convergence: If $\|\hat{\lambda} - \lambda\|_2 < \theta$ (a threshold), the algorithm converges. Otherwise, set $\lambda = \hat{\lambda}$ and go to step (1).

Note that in the Baum-Welch re-estimation formulas (10)-(13), $\gamma_t(j,k)$ can be considered as the posterior probability that, given the current model parameter λ, the observation o_t was generated from state S_j and accounted for by the k^{th} component of the Gaussian mixture density of state S_j.

3. THE BOOSTING BAUM-WELCH ALGORITHM

Ensemble methods for data analysis [12] that combine multiple models in a certain way have gained increased popularity in the last decade. They have proven to yield improved performance over the methods just using a single model in isolation on a range of tasks including classification and regression. Boosting [9] is an important instance of the ensemble methods. It involves training multiple models in sequence, one at an iteration. At each iteration, the error function designed for training the model is dependent of the performance of the model trained at the previous iteration on the training set. The theoretical study of Mason et al. [13] show that boosting can be viewed as a gradient descent search in a function space. Rosset et al. [14] apply the boosting framework to the problem of probability density estimation. Wang et al. [15] specialize it for the Gaussian mixture model (GMM). Tang et al. [16] extend this idea and apply the proposed Boosted-GMM algorithm to the problem of emotion recognition from speech and show promising recognition results. In this paper, we further extend the boosting framework to the HMM, and the resulting learning algorithm is named the boosting Baum-Welch algorithm.

Maximum likelihood estimation algorithms such as the Baum-Welch algorithm aim to find a local maximum of the HMM (log) likelihood function in order to determine the model parameters. Given an observation sequence, $O = o_1 o_2 \cdots o_T$, the HMM likelihood function is given by

$$p(O|\lambda) = \sum_{q_1 q_2 \cdots q_T} \left\{ \left[\prod_{t=1}^{T} b_{q_t}(o_t) \right] \left[\pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \right] \right\} \tag{14}$$

Apparently, the optimization of $p(O|\lambda)$ may consist of two alternating steps:

1. Assuming fixed initial state probabilities and state transition probabilities, the HMM likelihood function is a linear combination of the Gaussian mixture state emission densities. Maximizing the HMM likelihood function is equivalent to maximizing the individual likelihood functions of the Gaussian mixture state emission densities.

2. Assuming fixed Gaussian mixture state emission densities, the initial state probabilities and state transition probabilities are adjusted to maximize the HMM likelihood function.

The above procedures are performed iteratively until convergence. Each step is guaranteed to increase the HMM likelihood function, and convergence of the algorithm is achieved at a local maximum of the HMM likelihood function.

In this paper, we aim to robustly and accurately estimate the Gaussian mixture state emission densities of the HMM. Conventional techniques such as the Baum-Welch algorithm or the expectation maximization (EM) algorithm [17] are based on the optimization of the log likelihood function in the data space. Based on the ensemble framework for probability density estimation, the log likelihood function of the Gaussian mixture state emission densities can be optimized in a function space.

Suppose that we start with a Gaussian mixture density $f(o_t) = \sum_{k=1}^{M_1} c_k N(o_t|\mu_k, \Sigma_k)$. The log likelihood function of $f(o_t)$ is $L\{f(o_t)\} = \sum_{t=1}^{T} \log f(o_t)$. Consider adding to this Gaussian mixture density another Gaussian mixture density $g(o_t) = \sum_{k=M_1+1}^{M_2} c_k N(o_t|\mu_k, \Sigma_k)$ in a way such that

$$\hat{f}(o_t) = (1-\rho) f(o_t) + \rho g(o_t) \tag{15}$$

where ρ is a small positive number ($\rho \ll 1$) acting as an interpolation coefficient between the two Gaussian mixture densities $f(o_t)$ and $g(o_t)$. It is straightforward to show that $\hat{f}(o_t)$ is a valid Gaussian mixture density. The log likelihood function of $\hat{f}(o_t)$ is given by

$$\begin{aligned} L\left\{\hat{f}(o_t)\right\} &= L\{(1-\rho)f(o_t) + \rho g(o_t)\} \\ &= L\left\{(1-\rho)\left[f(o_t) + \frac{\rho}{1-\rho} g(o_t)\right]\right\} \\ &= T\log(1-\rho) + L\left\{f(o_t) + \frac{\rho}{1-\rho} g(o_t)\right\} \end{aligned} \tag{16}$$

Viewed as a function of f and g, $L\left\{f(o_t) + \frac{\rho}{1-\rho} g(o_t)\right\}$ is a function defined over a function space referred to as the Gaussian mixture density function space. Since $\rho \ll 1$, $\frac{\rho}{1-\rho} g(o_t)$ is small. A small perturbation of $f(o_t)$ in the function space will cause a change in L which can be well approximated by a first-order Taylor expansion of L around $f(o_t)$. That is,

$$L\left\{f(o_t) + \frac{\rho}{1-\rho} g(o_t)\right\} = L\{f(o_t)\} + \frac{\rho}{1-\rho} \sum_{t=1}^{T} \frac{g(o_t)}{f(o_t)} \tag{17}$$

Therefore, we have

$$L\left\{\hat{f}(o_t)\right\} = L\{f(o_t)\} + \left[T\log(1-\rho) + \frac{\rho}{1-\rho} \sum_{t=1}^{T} \frac{g(o_t)}{f(o_t)}\right] \tag{18}$$

Note that Equation (18) implies that in order to increase the likelihood function L, we need to guarantee that when we add a Gaussian mixture density to the existing Gaussian mixture density by Equation (15), the following inequality must hold

$$T\log(1-\rho) + \frac{\rho}{1-\rho} \sum_{t=1}^{T} > 0 \tag{19}$$

Therefore, we can perform the following two steps sequentially for each such addition

- For fixed ρ, maximize L with respect to g.
- For fixed g, maximize L with respect to ρ.

In order to maximize L for fixed ρ, we seek a g that maximizes $\sum_{t=1}^{T} \frac{g(o_t)}{f(o_t)}$. This goal can be accomplished by learning a Gaussian mixture density g from a sample of the observations $\{o_t\}_{t=1}^{T}$ weighted by $\frac{1}{f(o_t)}$. That is, more focus is given to those training examples which have low probabilities given by the density learned at the previous iteration. Intuitively, this is analogous to the boosting algorithm where the training examples mis-classified at the previous iteration are given bigger weights for the current iteration. Hence, when applied to HMM learning, the resulting algorithm is named the boosting Baum-Welch algorithm. The re-estimation formulas for the parameters of the Gaussian mixture state emission densities are revised accordingly, as follows:

$$\hat{c}_{jk} = \frac{\sum_{t=1}^{T} w_{jt}\gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{M} w_{jt}\gamma_t(j,k)} \tag{20}$$

$$\hat{\mu}_{jk} = \frac{\sum_{t=1}^{T} w_{jt}\gamma_t(j,k) o_t}{\sum_{t=1}^{T} w_{jt}\gamma_t(j,k)} \tag{21}$$

$$\hat{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} w_{jt}\gamma_t(j,k)(o_t - \hat{\mu}_{jk})(o_t - \hat{\mu}_{jk})^T}{\sum_{t=1}^{T} w_{jt}\gamma_t(j,k)} \tag{22}$$

where $1 \le j \le N, 1 \le k \le M, w_{jt} = \frac{1}{b_j(o_t)}$, and $\gamma_t(j,k)$ is given by Equation (13). The boosting Baum-Welch algorithm is summarized in Algorithm 1.

Algorithm 1 The boosting Baum-Welch algorithm

1: Input: $O = o_1 o_2 \cdots o_T$, L (number of boosting iterations).
2: Initialization: Π, A, B.
3: Repeatedly perform the following re-estimation procedures.
  3.1. Re-estimate the initial state probabilities and state transition probabilities using Equations (8) and (9).
  3.2. Set $b_j^{(0)}(o_t) = 0, w_{jt} = \frac{1}{T}, \quad 1 \le j \le N, 1 \le t \le T$.
  3.3. For $l = 1, 2, \cdots, L$, do
    3.3.1. Estimate $g_j^{(l)}(o_t)$ using Equations (20)-(22).
    3.3.2. Set $b_j^{(l)}(o_t) = (1-\rho) b_j^{(l-1)}(o_t) + \rho g_j^{(l)}(o_t)$
      – where $\rho = \arg\max_{0 \le \rho \le 1} L\left\{b_j^{(l)}(o_t)\right\}$
    3.3.3. Set $w_{jt} = \frac{1}{b_j^{(l)}(o_t)}$.
    3.3.4. If $L\left\{b_j^{(l)}(o_t)\right\} > L\left\{b_j^{(l-1)}(o_t)\right\}$ goto 3.3.1.
4: Output: final parameter estimates Π, A, B.

4. EXPERIMENTS

Speech emotion recognition is a relatively new direction in the areas of speech signal processing [18, 19, 20, 21, 22, 23]. It aims to extract the non-lexical, paralinguistic information from the speech signal regardless of its verbal content, and has turned out to be an important research topic with many useful practical applications.

In this paper, we perform speech emotion recognition experiments based on a speech database that we collected for a series of speech analysis and synthesis tasks. Our script consists of 720 semantically-neutral English sentences which were chosen to maximize the phonetic coverage of the English language [24]. A student actress whose mother language is American English was hired to speak each of these sentences, as naturally as possible, in the neutral, happy, sad, and angry manners, respectively. The speech waveforms were recorded in a studio environment at 44.1K Hz using a MOTU 8pre firewire audio interface and a Studio Projects B1 con-

Algorithm	Average Recognition Rate
Baum-Welch	87.8%
Boosting Baum-Welch	91.2%

Table 1. Speech emotion recognition experiment results.

denser microphone, and were downsampled to 16K Hz prior to further processing. The average length of the utterances in the database is about 3 to 4 seconds, depending on the emotion category.

For each experiment, we randomly selected from the database a training set consisting of 180 utterances per emotion and a test set consisting of 540 utterances per emotion. Therefore, the training set consists of $180 \times 4 = 720$ utterances in total and the test set $540 \times 4 = 2160$ utterances in total. Note that there are no overlapping utterances in the training and test sets. We ran 10 experiments independently, each of which involved a random selection of the training set and test set from the database, and the emotion recognition rates of these 10 experiments were averaged. We believe that in this way such average would represent a well generalized emotion recognition rate over the entire database. An experiment was carried out as follows. For each speech frame in an utterance, we extracted a set of basic acoustic features including 19 MFCCs (Mel-frequency cepstral coefficients), log energy, and pitch (f_0) using a 25ms hamming window at a 10ms frame rate. These basic acoustic features were augmented with their first and second derivatives to form a 63-dimensional feature vector.

For each emotion category, we trained an HMM using the Baum-Welch algorithm and the boosting Baum-Welch algorithm, respectively. The HMM was designed to have a left-to-right topology with 10 states, and each state is modeled by a Gaussian mixture state emission density. Maximum likelihood classification of the emotion was performed for every utterance in the test set.

We compare the average recognition results of the two HMM learning algorithms in Table 1. The experiment results clearly show that the proposed algorithm indeed improves the learning accuracy of the Gaussian mixture state emission densities of the emotion category dependent HMMs, leading to better emotion recognition performance than the original Baum-Welch algorithm.

5. CONCLUSION

In this paper, we propose a novel strategy for robustly and accurately learning the Gaussian mixture state emission densities of the HMM. The strategy is based on an ensemble or boosting framework for probability density estimation in which the learning of the Gaussian mixture densities is formulated as a gradient descent search in a function space. The resulting learning algorithm is named “the boosting Baum-Welch algorithm.” Our preliminary experiment results on emotion recognition from speech show that the proposed algorithm outperforms the original Baum-Welch algorithm on this task.

6. REFERENCES

[1] L.R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” Proceedings of the IEEE, 77 (2): 257–286, 1989.

[2] C.M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[3] L.R. Rabiner and B.H. Juang. Fundamentals of Speech Recognition. Prentice Hall, Upper Saddle River, N.J., 1993.

[4] C. Manning and H. Schutze. Foundations of Statistical Natural Language Processing. MIT Press. Cambridge, MA: May 1999.

[5] J. Hu, M. Brown, W. Turin, “HMM Based On-Line Handwriting Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(10):1039–1045, 1996.

[6] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” Ann. Math. Statist., 41(1):164–171, 1970.

[7] G.J. McLachlan, D. Peel. Finite Mixture Models. Wiley, 2000.

[8] R.O. Duda, P.E. Hart, D.G. Stork. Pattern classification (2nd edition). Wiley, New York, 2001.

[9] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” Journal of Computer and System Sciences, 55(1):119–139, 1997.

[10] B. Schuller, S. Steidl, A. Batliner, “The Interspeech 2009 Emotion Challenge,” Interspeech, Brighton, UK, 2009.

[11] D. Povey, S. Chu, B. Varadarajan, “Universal Background Model Based Speech Recognition,” ICASSP 2008.

[12] R. Berk, “An Introduction to Ensemble Methods for Data Analysis,” Department of Statistics Papers, UCLA, 2005.

[13] L. Mason, J. Baxter, P. Bartlett, M. Frean, “Boosting algorithms as gradient descent,” NIPS 1999.

[14] S. Rosset, E. Segal, “Boosting density estimation,” NIPS 2002.

[15] F. Wang, C. Zhang and N. Lu. Boosting GMM and its two applications. MCS 2005.

[16] H. Tang, S. Chu, M. Hasegawa-Johnson, T. Huang. “Emotion Recognition from Speech via Boosted Gaussian Mixture Models,” ICME 2009.

[17] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm.” J. of Royal Stat. Society, B, 39(1):1–38, 1977.

[18] D. Seppi, A. Batliner, B. Schuller, S. Steidl, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, N. Amir, and V. Aharonson, “Patterns, Prototypes, Performance: Classifying Emotional User States,” Interspeech, Brisbane, Australia, 2008.

[19] D. Ververidis, C. Kotropoulos, “Emotional speech recognition: resources, features, and methods,” Speech Communication 48(9):1162-1181, 2006.

[20] D. Neiberg, K. Elenius, and K. Laskowski, “Emotion recognition in spontaneous speech using GMMs,” Interspeech 2006.

[21] C. Lee, S. Yildirim, M. Bulut, A. Kazemzadeh, C. Busso, Z. Deng, S. Lee, S. Narayanan, ”Emotion recognition based on phoneme classes,” ICSLP 2004.

[22] T.L. Nwe, S.W. Foo and L.C. De Silva. Speech emotion recognition using hidden markov models. Speech Communication 41, pp. 603–23, 2003.

[23] H. Tang and T. Huang, “Boosting Gaussian Mixture Models via Discriminant Analysis,” 2008 IEEE International Conference on Pattern Recognition, Tempa, FL, December, 2008.

[24] Harvard Sentences: From the appendix of: IEEE Subcommittee on Subjective Measurements IEEE Recommended Practices for Speech Quality Measurements. IEEE Transactions on Audio and Electroacoustics. vol 17, 227–46, 1969.